REMARKS

Claims 1-10, 13-21 and 25 were examined in the Final Office Action mailed on 11/13/2008 (hereafter "Outstanding Final Office Action"). All the claims were rejected.

Reconsideration and withdrawal of the final rejection is respectfully requested on the grounds that there is at least one previously presented independent claim that is allowable over the art of record, as explained in the following remarks.

Claim Rejections - 35 USC § 112, First Paragraph

Claims 7 and 10 are rejected under 35 U.S.C. § 112, first paragraph, allegedly as failing to comply with the written description requirement. It was stated that the claim(s) contains subject matter which was not described in the specification in such a way as to reasonably convey to one skilled in the relevant art that the inventor(s), at the time the application was filed, had possession of the claimed invention. In particular, it was alleged that the terms "provides said identifier to said user program" and "providing said identifier to said user program" are new matter.

In response, it is respectfully noted that the drawings as filed is part of the application as filed, and the above two terms are supported by line 209 of Figure 2, in combination with paragraph 39 of the specification, which states:

> [0039] Lines 207 through 209 are respectively shown containing declarations Typedef int (*fptr)(); (fptr is a pointer type variable, declared as an integer), Tstep(int, fptr, fptr) (accepts three parameters respectively of type integer, pointer, and a pointer), and **int txid = Tinit() (Tinit() returns a transaction identifier which is assigned to an integer type variable txid).**
>
> ...
>
> [0051] Program interface block 310 may provide an interface using which **programs may interact with the transaction manager** to execute atomic transactions according to various aspects of the present invention. Program interface block 310 may receive appropriate data in response to execution of procedure calls (four, in the illustrative example described above), and forwards the received data to transaction management block 330. **In case of transactions such as Tinit(), the transaction identifier is passed back to the execution environment (which executes software instructions).** (**Emphasis Added**)

One skilled in the relevant arts will clearly appreciate from the above noted/ emphasized portion that the claimed (transaction) identifier generated by **the Tinit() (part of the transaction manager) is returned, and thus provided to the user program**.

Withdrawal of the rejections under 35 U.S.C. § 112 is respectfully requested.

Claim Rejections - 35 USC § 103

Claims 1-2, 5-10, 13, 16-17, 20-21 and 25 were rejected under 35 U.S.C. § 103(a) as being unpatentable over Applicant's Admitted Prior Art, Fig. 1 and specification paragraphs 22-33 ("AAPA"), in view of Gostanian *et al*, U.S. 5,781,910 (hereafter "Gostanian"), and in view of Lordi *et al*, U.S. 5,857,204 (hereafter "Lordi").

Applicants respectfully traverse and request withdrawal of the final rejection. The broad differences are first pointed out for the convenience of the Examiner and also to further prosecution.

Broad Differences from Art of Record

It is believed that there are two classes of prior art in the art of record:

C1. As in the Applicant's admitted prior art of Figure 1 of the subject patent application, the user applications being required to keep track of the specific task procedures (of an atomic transaction) and to execute the corresponding roll back procedures in a reverse order (in case of aborting the atomic transaction in the middle), with the **program logic of the roll back procedures also being specified by the programmer/user in the user applications**; and

C2. A common operating system, **transparent to the user applications**, has the program logic (including the roll back procedures) for undoing the effects of task procedures in case an atomic transaction is to be aborted in the middle. Both Gostanian and Lordi are believed to fall in this class.

The prior art of C1 would be onerous to the programmers of user applications, while the prior art of C2 does not provide sufficient control (in terms of specifying any desired custom logic for the rollback procedures) to individual programmers of user applications.

The technology of the subject patent application addresses the problems of C1 and C2 by using a novel approach, which is not taught or reasonably suggested by the two classes (either individually or in combination).

In particular, the underlying transaction manager of the subject patent application is used by user applications so that the programmer/user of the user programs is relieved of the burden of keeping track of the specific task procedures executed. In addition, the programmer is provided the ability to specify desired custom logic.

As explained first below, each reference of record is deficient in even in broad aspects.

AAPA (Applicant's admitted prior art) is deficient in that the programmer of the user program has to mentally keep track of the task procedures at each potential abort points (i.e., by the time execution reaches if conditions of line 120 and 160 of Figure 1) while coding the user program.

Gostanian is deficient in that the rollback logic is entirely provided by the shared manager process, and thus the programmer/user of user programs does not have the ability to specify any desired custom logic for the rollback logic.

Lordi does **not** relate to implementation of **atomic transactions in user programs**! Rather the teachings relate to applying transaction techniques to file system type operations, which are **transparent at the user program level**. In other words, the user specify operations such as writing, rename, etc., (which would be at the user program level), and the teachings there would use transaction techniques to provide atomicity in the execution of the specified operations. There is no disclosure or suggestion in Lordi that the perform, finalize and undo type routines (Col. 3 lines 45-55 of Lordi) can be used in complex **data processing applications (which would be akin to the claimed user programs**).

Irrespective, there are several deficiencies in the way the Examiner compares the claims with the disclosure of the references, as explained below.

Erroneous Comparisons

With respect to independent claim 7, the user program requests an identifier, which is generated and **provided by the transaction manager to the requesting user program**. A variable in the user program is then set equal to the provided identifier. This variable is used in each combination (in the user program), which further specifies a task procedure and the corresponding rollback procedure.

None of the references of record teach or reasonably suggest such a sequence of features.

In particular, APAA teaches neither a request for the identifier by the user program nor setting a variable in the user program. Further, there is no disclosure or suggestion that the 'variable' is used in each of the combinations. In particular, the "Account1()" (sought to be equated by the Examiner to the claimed identifier) **is not set to any variable in Figure 1** or corresponding disclosure.

The transaction identification code (sought to be equated by the Examiner to the claimed identifier) of Gostanian is also **clearly not provided to the user programs**. See, for example, Figure 5 of Gostanian, which shows request 510 being received from application client (AC) arguably akin to the claimed user program, and reply (to client) 574 being again sent to AC. There is no teaching or suggestion in Gostanian that reply 574 contains the transaction identifier.

Lordi also does not cure that deficiency since the teachings with respect to atomic transaction are not in user programs, as explained above.

Thus, it is concluded that none of the references of record teaches or reasonably suggests one or more features of independent claim 7.

Independent claim 7 is allowable over the art of record for other reasons. For example, claim 7 recites that the **abort operation be specified in the user program using the identifier** (associated with an abort procedure) provided earlier by the transaction manager.

The Examiner relies on Lordi for such a feature (see Page 11 last 6 lines of Outstanding Final Office Action). For reasons noted above, Lordi does not contemplate implementing logic underlying atomic transactions in user programs. Accordingly, the abort operation would not be specified in the user program. Furthermore, it is noted that the portions relied upon (Col. 2 line 66 to Col. line 7 of Lordi) by the Examiner discloses that the abort operation is specified within a database system, and not specified in an application (user program). See Col. 2 lines 19-21 of Lordi.

The Examiner concedes AAPA does not teach the above-quoted feature (see page 11, first 2 lines of the last paragraph of the Outstanding Final Office Action).

Gostanian also would not have such a feature since the decision to abort is performed within a coordinator (see Figures 4 and 5 of Gostanian), which is not part of the application client (arguably analogous to the claimed user programs).

Furthermore, claim 7 recites that the identifier is unique and thus cannot be equated to Account1(), contrary to the allegation of the Examiner.

Independent claim 7 is allowable over the art of record for those reasons as well. The remaining presented independent claims are also allowable over the art of record at least for one of the reasons noted above. The dependent claims are allowable at least for the reasons noted above with respect to the corresponding base claim.

Impermissible Hindsight Employed

Applicants further note that it is a long felt general need to provide enhanced flexibility in implementing custom atomic transactions during the **development of user programs**. The present invention provides such a flexibility, possibly based on several concepts which exist individually in different areas and/or prior solutions. However, any alleged combination of such prior concepts along the lines of the claimed inventions, is simply based on impermissible hindsight gleaned from Applicant's disclosure.

The Examiner impermissibly uses the claims as a template to select features from the two classes of prior art (C1 and C2) in alleging that the references of record render obvious the claimed inventions.

As evidence of such a conclusion, it is first pointed out that the combination of advantages of the present invention (noted above) are new and not obtained/yielded by the references of record, even together. This advantage is clear indication that the present invention is not rendered obvious by the art of record.

The Examiner is respectfully reminded of the burden under MPEP Section "707.07(f) Answer All Material Traversed" to answer the advantages/arguments thus presented.

It is further pointed out that the several specific combination of features relied upon by the Examiner would not work together. For example, AAPA relies on hardcoded (i.e., those coded by the programmer) software instructions to keep track of the rollback procedures, while Lordi relies on a log file for undoing the effects of performance of file operations. These two techniques are fundamentally incompatible and thus the alleged combination is asserted to be inoperative, and thus not a valid combination in rendering obvious the presented independent claims.

Conclusion

Thus, it is believed that all objections and rejections have been overcome and continuation of examination is respectfully requested. The Examiner is invited to telephone the undersigned representative at 707.356.4172 if it is believed that an interview might be useful for any reason.

Respectfully submitted,
/Narendra Reddy Thappeta/
Signature
Printed Name: Narendra Reddy Thappeta
Attorney for Applicant
Registration Number: 41,416

Date: 29 June 2009